<u>AMENDMENTS TO THE CLAIMS:</u>

This listing of claims replaces all prior versions and listings of the claims in the present application:

<u>LISTING OF CLAIMS:</u>

1.   (Currently Amended) A method for avoiding section collision for application server requests over a single database connection <u>between a database server and an application server</u>, the method comprising:

(a)   receiving a first statement assigned a first command source identifier by ~~a~~ <u>the</u> database server from a first application source over ~~a~~ <u>the</u> single database connection ~~between the database server and an application server~~, <u>the first command source identifier uniquely identifying an application source to the first statement</u>;

(b)   receiving a second statement assigned a second command source identifier by the database server from a second application source over the single database connection, wherein the first statement is substantially identical to the second statement; <u>the second command source identifier uniquely identifying an application source to the second statement</u> and<u>,</u>

(c)   executing the first statement assigned a first command source identifier separately from and in parallel with the second statement assigned the second command source without section collision.

2.   (Original) The method of claim 1, wherein the first and second command source identifiers are assigned by the application server.

3.   (Original) The method of claim 1, wherein the first application source and the second application source are different application sources within a same application.

4.   (Currently Amended) The method of claim 1, wherein the first statement assigned the first command source identifier and the second statement assigned the second command source identifier is are combined into a single statement containing the first and second application sources, wherein a portion of the single statement pertaining to the first application source is assigned the first command source identifier, and wherein a portion of the single statement pertaining to the second application source is assigned the second command source identifier.

5.   (Original) The method of claim 1, wherein the first application source is within a first application and the second application source is within a second application.

6.   (Currently Amended) The method of claim 1, wherein the first statement assigned the first command source identifier is a statement to open a cursor and wherein the second statement assigned the second command source identifier is a statement to open the same cursor.

7.   (Currently Amended) The method of claim 6, wherein the executing (c) further comprises:

(c1)   creating a first instance of the cursor in response to the first statement assigned the first command source identifier;

(c2)   assigning a first query identifier to the first instance of the

cursor by the database server;

(c3) creating a second instance of the cursor in response to the second statement assigned the second command source identifier;

(c4) assigning a second query identifier to the second instance of the cursor by the database server; and

(c5) returning the first ~~query identifier~~ and second query ~~identifier~~ identifiers to the application server.

8. (Currently Amended) A method for avoiding section collision for application server requests over a single database connection, the method comprising:

(a) receiving a first statement to open a cursor by a database server over a single database connection between the database server and an application server;

(b) creating a first instance of an ~~the~~ open cursor in response to the first statement;

(c) assigning the first instance a first query identifier, the first query identifier uniquely identifies the first instance of the open cursor;

(d) receiving a second statement to open the ~~same~~ cursor by the database server over the single database connection before the first instance of the open cursor closes;

(e) creating a second instance of the open cursor in response to the second statement; and

(f) assigning the second instance a second query identifier, the second query identifier uniquely identifies the second instance of the open cursor.

9. (Original) The method of claim 8, wherein the first and second query identifiers are assigned by the database server.

10.   (Original) The method of claim 8, further comprising:

(g)   processing the first instance of the cursor separately from and in parallel with the second instance of the cursor.

11.   (Original) The method of claim 8, further comprising:

(g)   returning the first and second query identifiers to the application server.

12.   (Currently Amended) The method of claim 11, wherein subsequent statements received by the database server for the first instance of the cursor ~~comprises~~ include the first query identifier.

13.   (Currently Amended) The method of claim 11, wherein subsequent statements received by the database server for the second instance of the cursor ~~comprises~~ include the second query identifier.

14.   (Currently Amended) The method of claim 8, wherein the first statement is from a first application source and is assigned a first command source identifier, <u>and</u> ~~wherein~~ the second statement is from a second application source and is assigned a second command source identifier.

15-28. (Canceled)